GET: FDR envelopes

Mari Myllymäki
Natural Resources Institute Finland (Luke)

Abstract

This vignette gives examples of the use of false discoverate rate (FDR) envelopes that are proposed in (Mrkvička and Myllymäki 2023) and implemented in the R package **GET**. When citing the vignette please cite Mrkvička and Myllymäki (2023) and Myllymäki and Mrkvička (2020, **GET**: Global envelopes in R. arXiv:1911.06583 [stat.ME]) available at https://arxiv.org/abs/1911.06583.

Keywords: false discovery rate, functional linear model, global envelope test, local spatial correlation, multiple testing, resampling, R.

1. Introduction

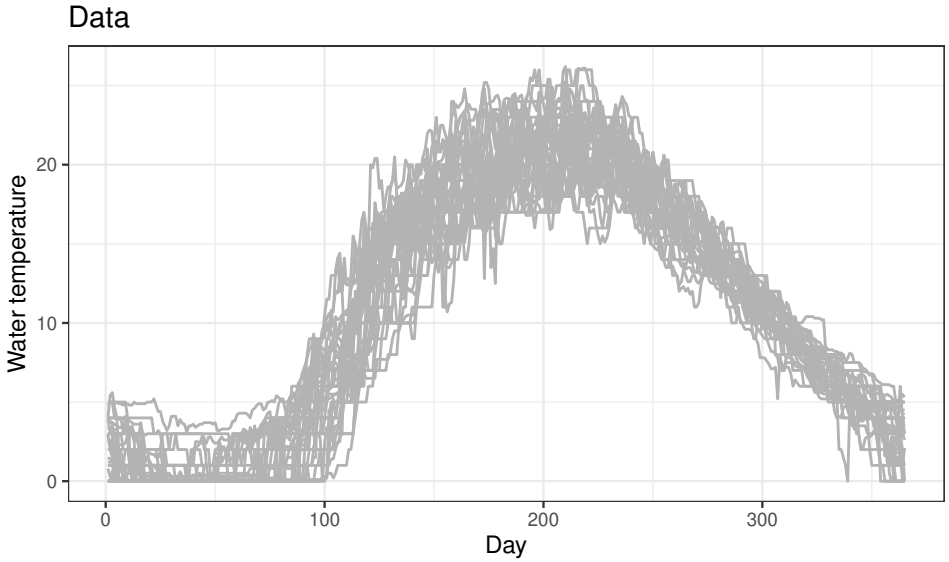
This vignette gives examples of the use of false discovery rate (FDR) envelopes, which are implemented in the R (R Core Team 2020) package **GET**. Three examples given below correspond to those presented in Mrkvička and Myllymäki (2023). The envelope plots are produced by the use of the **ggplot2** package (Wickham 2016), where we utilize the theme `theme_bw` for this document, and also the **patchwork** package (Pedersen 2020).

```
R> library("GET")
R> library("ggplot2")
R> theme_set(theme_bw(base_size = 9))
R> library("patchwork")
R> library("readxl")
R> library(patchwork)
R> library(spatstat)
```

2. Annual water temperature curves

The data of annual water temperature curves sampled at the water level of Rimov reservoir in the Czech republic every day from 1979 to 2014 are available as the data `rimov` in the **GET** package.

```
R> data("rimov")
R> plot(rimov) + labs(x="Day", y="Water temperature", title="Data")
```



Viewing the observations as daily samples of functional data, Mrkvička and Myllymäki (2023) fit the model

$$y_i(k) = \beta_0(k) + \beta_1(k)(i - 1978) + e(k), \tag{1}$$

where $y_i(k)$ is the water temperature at day k, $k = 1, \ldots, 365$, in year i, $i = 1979, \ldots, 2014$, $\beta_0(k)$ and $\beta_1(k)$ are model parameters, and $e(k)$ denotes the error. To test the null hypothesis

$$\beta_1(k) = 0 \quad \text{for all } k = 1, \ldots, 365, \tag{2}$$

they took the estimator of the regression coefficient, i.e. $\hat{\beta}_1(k)$, as the test statistic following the methodology proposed in Mrkvička, Roskovec, and Rost (2021).

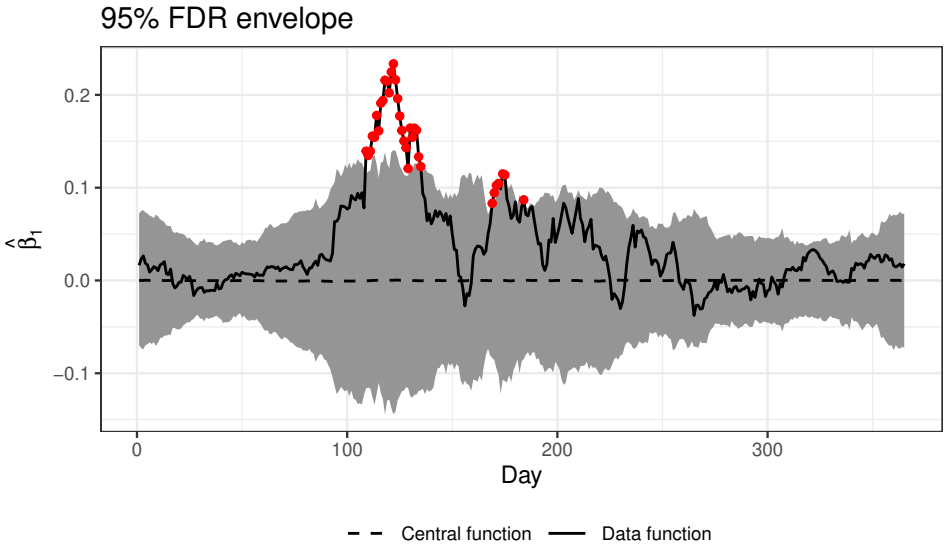
The function `graph.flm()` of **GET** allows the use of tests in functional general linear models based on regression coefficients. The simulations under the null hypotheses are performed using the Freedman and Lane (1983) procedure and, for the multiple testing correction due to the use of multivariate (functional) test statistic, the function `graph.flm()` provides both control of FDR (local test) or family-wise error rate (FWER, global test). The multiple testing control is chosen by the argument `typeone`. If `typeone = "fdr"`, the the FDR envelopes proposed by Mrkvička and Myllymäki (2023) are used. The choice `typeone = "fwer"` leads to the use of global envelope tests, see `global_envelope_test()`.

Thus, the test of null hypothesis (2) using the $\hat{\beta}_1(k)$ as the test statistic and the FDR control can be performed as follows. Here `nsim` specifies the number of permutations of the Freedman and Lane (1983) method, `formula.full` specifies the full model, `formula.reduced` specifies the reduced model where the interesting factor whose significance is to be tested is dropped out in comparison to the `formula.full`, and the data are provided in `curve_sets` and `factors`.

```
R> nsim <- 7000
R> res <- graph.flm(nsim=nsim,
+    formula.full = Y~Year,
+    formula.reduced = Y~1,
+    typeone = "fdr",
+    curve_sets = list(Y=rimov),
+    factors = data.frame(Year = 1979:2014-1978))
```

The result is plotted using the `plot()` function; because the plotting of **GET** utilizes **ggplot2** for plotting, e.g., labels can be changed as shown below using the **ggplot2** style.

```
R> plot(res) +
+    labs(x="Day", y=expression(hat(beta)[1]))
```



Increase (positive β-coefficient) is observed in some the spring and summer days.

3. Population growth example

The data

```
R> data("popgrowthmillion")
```

contains population growth, i.e. population at the end of the year divided by population at the beginning of the year, in 134 countries in years from 1950 to 2015. The dataset includes only countries over million inhabitants in 1950. The data were extracted from the supplement of Nagy, Gijbels, and Hlubinka (2017) distributed under the GPL-2 license.

For the population data available in **GET**, the continents/groups of the countries are as follows and we make a `curve_set` object of the data for nicer treatment of it. The Oceania group with three countries was left out from the analysis.

```
R> POPcset <- create_curve_set(list(r=1950:2014,
+                 obs=popgrowthmillion[, -(132:134)])) # Left out Oceania (Australia, New Z
R> groups <- c(rep("Africa", times=37),
+               rep("Asia", times=37),
+               rep("Europe and North America", times=35),
+               rep("Latin America", times=20),
+               rep("Europe and North America", times=2))
```

We further use the GDP data which was obtained from World Bank and prepared as shown in Appendix A, and which is available at **GET** as a data object `GDP`.

```
R> data("GDP")
R> GDPcset <- GDP
```

We only consider those countries that exist in both data and check that the data sets are arranged similarly.

```
R> #- Reduce POPcset to those countries that exist in GDP data
R> cond <- colnames(POPcset$funcs) %in% colnames(GDPcset$funcs)
R> POPcset <- subset(POPcset, cond)
R> groups <- groups[cond]
R> #- Check that the GDPcset has the same order as the POPcset and groups!
R> if(any(colnames(GDPcset$funcs) != colnames(POPcset$funcs)))
+    stop("Order of countries do not match.")
R> #- Reduce POPcset (and GDPcset) to years 1960-2014
R> POPcset <- crop_curves(POPcset, r_min =1960, r_max=2014)
R> GDPcset <- crop_curves(GDPcset, r_min =1960, r_max=2014)
R> groups <- as.factor(groups)
```

Thus, we have the group (Continent), population growth curves and GDP curves 1960-2014:

```
R> table(groups)
```

```
groups
                  Africa                     Asia
                      34                       28
Europe and North America            Latin America
                      34                       18
```

```
R> POPcset
```

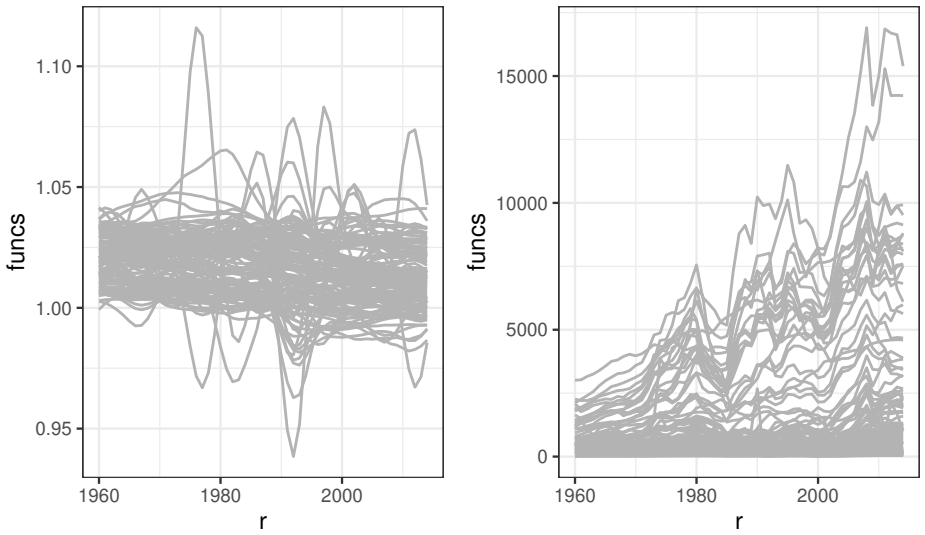
```
A curve_set(1d) object with 114 curves observed at 55 argument values.
Contains:
$ r     :  int [1:55] 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 ...
$ funcs :  num [1:55, 1:114] 1.02 1.02 1.02 1.02 1.02 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:55] "1960" "1961" "1962" "1963" ...
  ..$ : chr [1:114] "Burundi" "Eritrea" "Ethiopia" "Kenya" ...
```

```
R> GDPcset
```

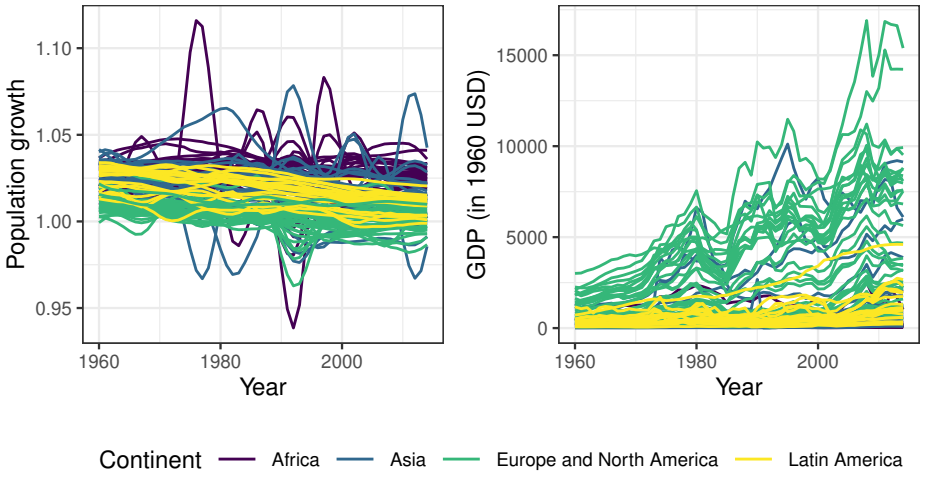
```
A curve_set(1d) object with 114 curves observed at 55 argument values.
Contains:
$ r     :  int [1:55] 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 ...
$ funcs :  num [1:55, 1:114] 70.1 70.2 71.6 75.7 81.8 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:55] "1960" "1961" "1962" "1963" ...
  ..$ : chr [1:114] "Burundi" "Eritrea" "Ethiopia" "Kenya" ...
```

Plotting the data simply

```
R> p1 <- plot(POPcset)
R> p2 <- plot(GDPcset)
R> p1 + p2
```



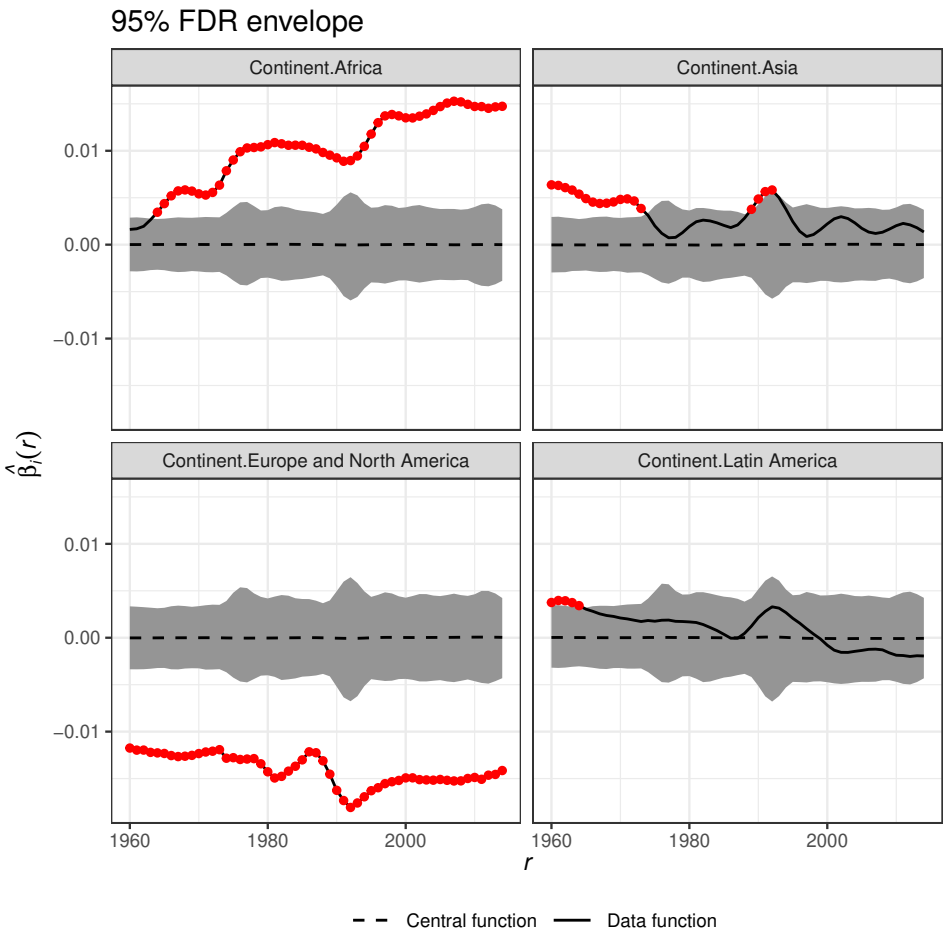
The following plot further shows the different continents with different colors.



Mrkvička and Myllymäki (2023) explored the functional general linear model (GLM) for the population growth across years 1960-2014 with categorical covariate (continent), continuous covariate (GDP) and interactions (GDP with respect to the continent). For this purpose, they applied three tests, and on each of them, they used their proposed FDR control (IATSE) to obtain all years which are significant for the studied covariate. We refer the reader to Mrkvička and Myllymäki (2023) about the details and show below how these three tests can be run using **GET**.

First the effect of continent in the main effects model (formula.full) was tested. The function `graph.flm()` was used to obtain simulations under the null and to test the effect under the control of FDR. The result can be directly plotted. The functional dependent variable (Y) as well as the functional explanatory variable (GDP) are provided as a `curve_set` objects in the argument `curve_sets`, while the factors (here Continent) that are constant for the dependent function can be provided in a data frame in the argument `factors`.

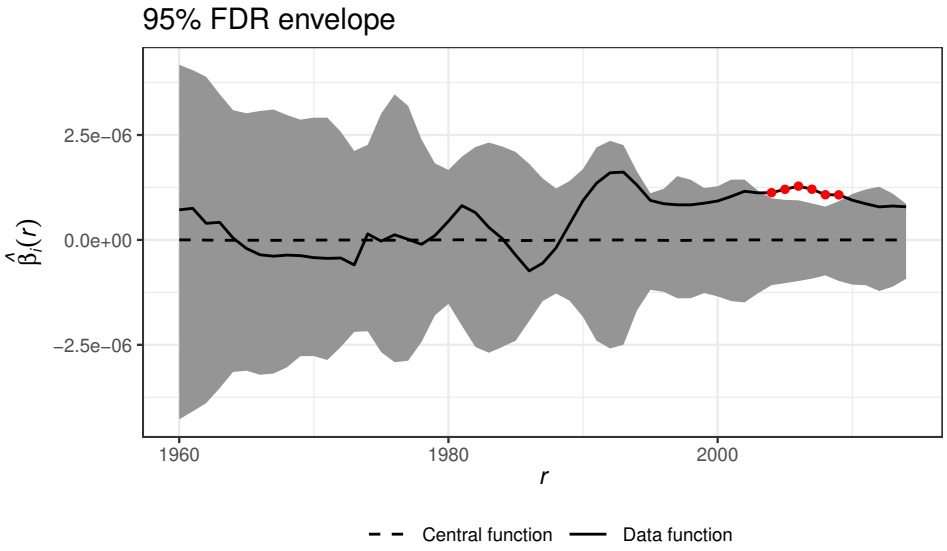
```
R> nsim <- 5000
R> resC <- graph.flm(nsim = nsim,
+                    formula.full = Y ~ GDP + Continent,
+                    formula.reduced = Y ~ GDP,
+                    typeone = "fdr", # Choose the control
+                    curve_sets = list(Y=POPcset, GDP=GDPcset),
+                    factors = data.frame(Continent = groups),
+                    contrasts = FALSE)
R> plot(resC)
```



For nicer visualization, Mrkvička and Myllymäki (2023) further added the fitted curve for every continent, i.e. the intercept β_i^0 and mean effect of GDP, $\beta_i^{\mathrm{GDP}} \cdot \mathrm{mean}(\mathrm{GDP}_i)$, were added to every $\beta_{j,i}^{\mathrm{Cont}}$.

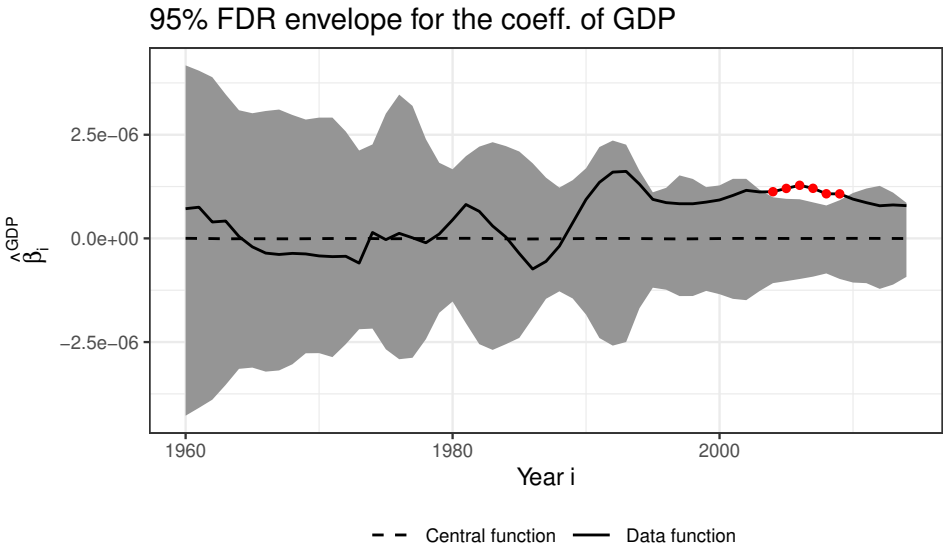
Second, the effect of GDP was tested:

```
R> resG <- graph.flm(nsim = nsim,
+                    formula.full = Y ~ GDP + Continent,
+                    formula.reduced = Y ~ Continent,
+                    typeone = "fdr", # Choose the control
+                    curve_sets = list(Y=POPcset, GDP=GDPcset),
+                    factors = data.frame(Continent = groups),
+                    contrasts = FALSE)
R> # Basic plot:
R> plot(resG)
```



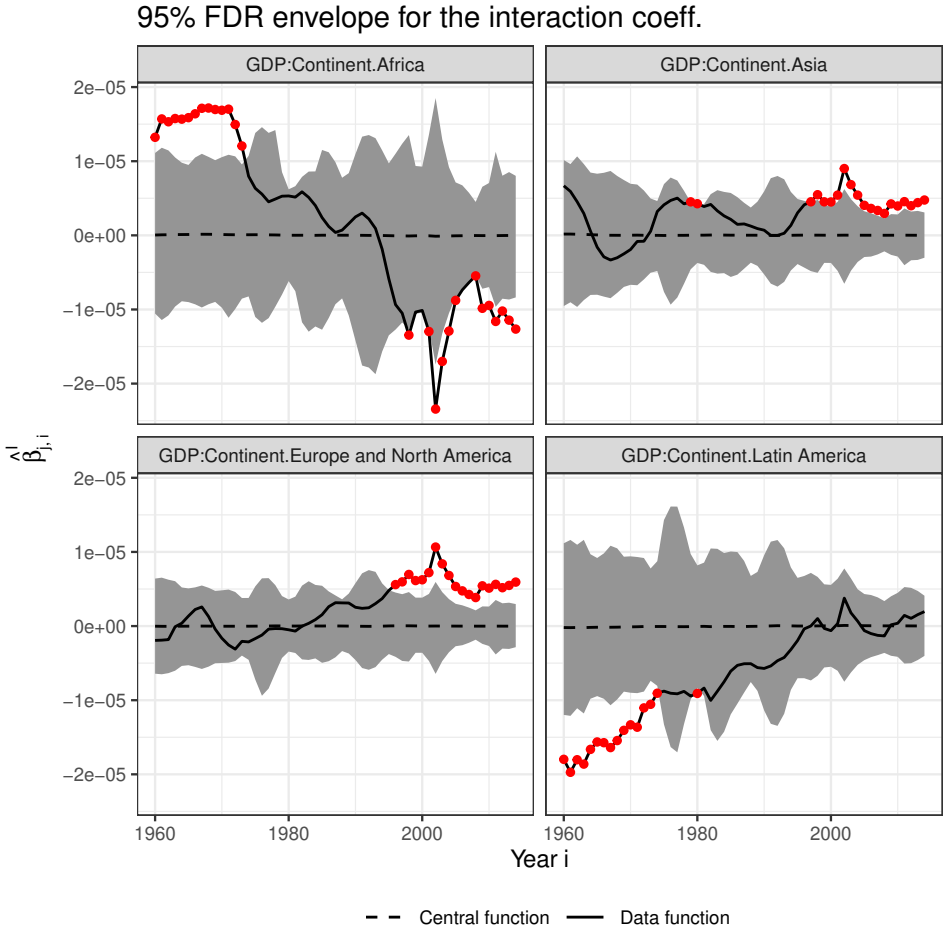
Here we again illustrate that the labels of the default envelope plots can be tuned rather easily:

```
R> labTextx <- 'Year'
R> plot(resG) +
+    labs(x=bquote(.(labTextx[1]) ~ i),
+         y=expression(hat(beta)[i]^GDP),
+         title="95% FDR envelope for the coeff. of GDP")
```



Third, the effect of the interaction of continent and GDP was tested:

```
R> resCG <- graph.flm(nsim = nsim,
+                     formula.full = Y ~ GDP + Continent + GDP:Continent,
+                     formula.reduced = Y ~ GDP + Continent,
+                     typeone = "fdr",
+                     curve_sets = list(Y=POPcset, GDP=GDPcset),
+                     factors = data.frame(Continent = groups),
+                     contrasts = FALSE, savefuns = TRUE)
R> # Tuned labels
R> plot(resCG) +
+    labs(x=bquote(.(labTextx[1]) ~ i),
+         y=expression(hat(beta)[list(j,i)]^I),
+         title="95% FDR envelope for the interaction coeff.")
```



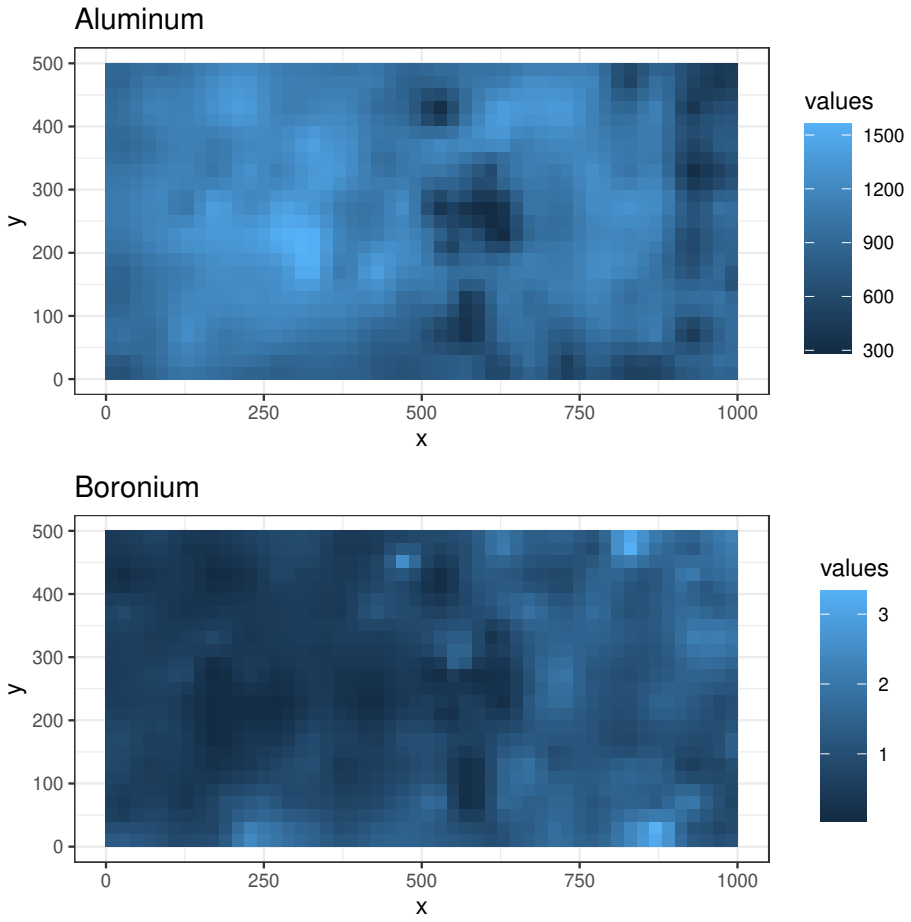
4. Local spatial correlation

The soil data in a rain forest plot used also in Mrkvička and Myllymäki (2023) are openly available through `http://ctfs.si.edu/webatlas/datasets/bci/soilmaps/BCIsoil.html`. Please note the copyright and acknowledgement information at the webpage.

The soil data are observed in the tropical forest plot of area 1000 m × 500 m in Barro Colorado Island (Hubbell, Condit, and Foster 2005; Hubbell, Foster, O'Brien, Harms, Condit, Wechsler, Wright, and Loo de Lao 1999; Condit 1998). This data set is accompanied with many covariates but here we study only the dependence of the first two covariates, i.e., Aluminium (Al) and Boronium (B) observed on a 50×25 grid with equally spaced observations.

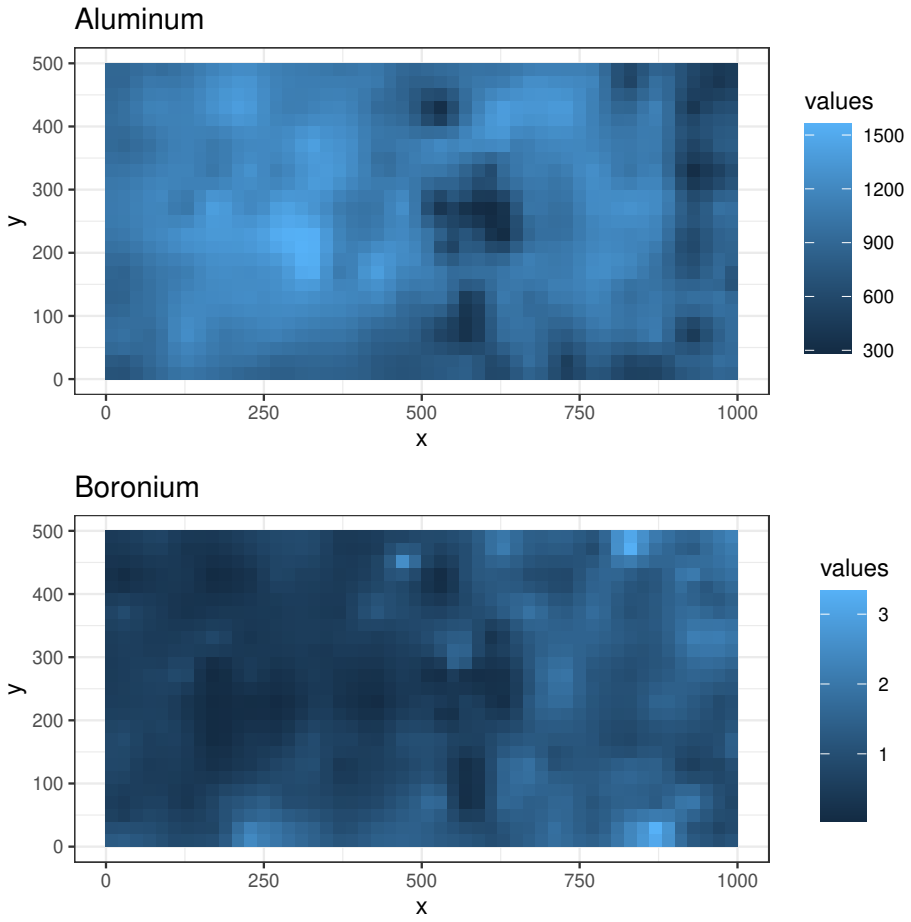
The data we used was available as an excel file at `http://ctfs.si.edu/webatlas/datasets/bci/soilmaps/bci.block20.data.xls`. After downloading the file, it can be read it to R

```
R> library(openxlsx)
R> data <- read_excel("bci.block20.data.xls", sheet=2)
```



Let us then first plot the data using **ggplot**. Also preparation of the data to data frames.

```
R> data <- as.data.frame(data) # Turn into a normal data.frame
R> df1 <- data.frame(x=data[,"x"], y=data[,"y"], values=data[,"Al"],
+                    Variable="Aluminum")
R> df2 <- data.frame(x=data[,"x"], y=data[,"y"], values=data[,"B"],
+                    Variable="Boronium")
R> p1 <- ggplot() +
+    GET:::geom_raster_fixed(data=df1, aes(x=.data$x, y=.data$y,
+                                          fill=.data$values),
+                            width=20, height=20) +
+    coord_fixed() + ggtitle("Aluminum")
R> p2 <- ggplot() +
+    GET:::geom_raster_fixed(data=df2, aes(x=.data$x, y=.data$y,
+                                          fill=.data$values),
+                            width=20, height=20) +
+    coord_fixed() + ggtitle("Boronium")
R> p1 + p2 + plot_layout(ncol=1)
```



The dataframes `df1` and `df2` contain our data now. Let's put them into a joint data frame, where the first two columns correspond to the values of the two random fields whose correlations are to be studied, and the third and fourth columns correspond to the x- and y-coordinates where these random fields have been observed. Further, width and height gives the sizes of the grid cells that the x- and y-locations represent (here of equal size).

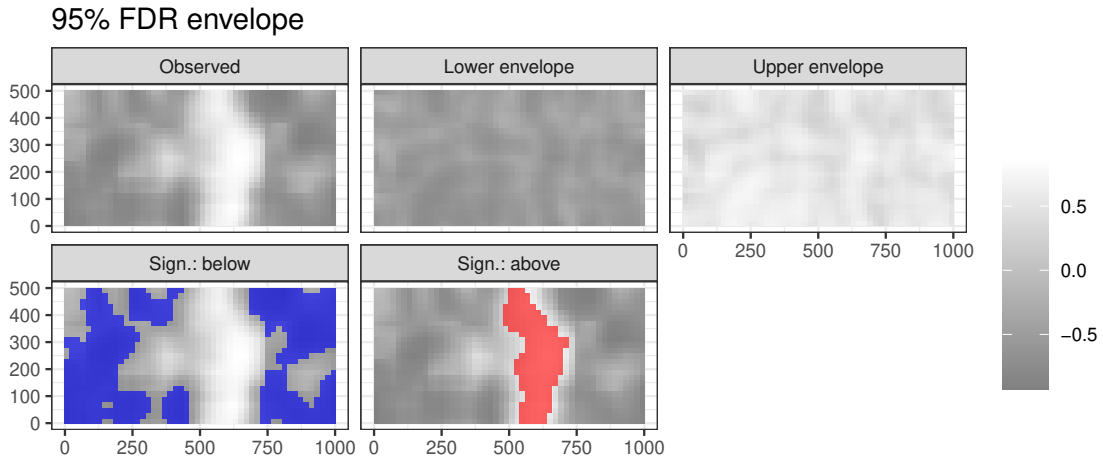
```
R> # Observations should be at the same locations.
R> # (Otherwise some smoothing methods must be applied first
R> # to get values on the same coordinates.)
R> if(!identical(df1$x, df2$x) || !identical(df1$y, df2$y))
+    message("Data observed on different locations!")
R> Var1 <- df1[!is.na(df1$values) & !is.na(df2$values),]
R> Var2 <- df2[!is.na(df1$values) & !is.na(df2$values),]
R> dat <- data.frame(Var1 = Var1$values, Var2 = Var2$values,
+                    X = Var1$x, Y = Var1$y,
+                    width = 20, height = 20)
R> head(dat)

      Var1      Var2  X   Y width height
1 700.1829 1.2898131 10  10    20     20
```

```
2 767.6347 0.9908329 10  30     20      20
3 908.5443 0.6311114 10  50     20      20
4 963.0225 0.5451379 10  70     20      20
5 944.6335 0.6036719 10  90     20      20
6 887.1651 0.7255532 10 110     20      20
```

Viladomat, Mazumder, McInturff, McCauley, and Hastie (2014) procedure requires a smoothing parameter `Delta` of the local correlation. We use a range of values according to Viladomat *et al.* (2014).

Then we can study the local correlations controlling for the false discovery rate (FDR, chosen in the argument `typeone`) as follows:

```
R> Delta <- seq(0.1,0.9,0.1)
R> # nsim = 1000 runs for about 10 minutes (with one kernel)
R> res <- GET.localcor(data = dat, Delta = Delta, nsim = 1000, typeone = "fdr",
+                      varying.bandwidth = FALSE, bandwidth.h = 100)
R> plot(res, sign.type = "col", what = c("obs", "lo", "hi", "lo.sign", "hi.sign"))
```



The output of the test shows us that there are significant negative correlations at the left and right sides of the region, and positive correlations in the middle of the region.

Acknowledgements

The soil data comes from the BCI forest dynamics research project that was founded by S.P. Hubbell and R.B. Foster and is now managed by R. Condit, S. Lao, and R. Perez under the Center for Tropical Forest Science and the Smithsonian Tropical Research in Panama. Numerous organizations have provided funding, principally the U.S. National Science Foundation and hundreds of field workers have contributed. The Barro Colorado Island soils data set was collected and analyzed by J. Dalling, R. John, K. Harms, R. Stallard, and J. Yavitt, with support from National Science Foundation, grants DEB021104, DEB021115, DEB0212284, DEB0212818 and OISE 0314581, Smithsonian Tropical Research Institute and Center for Tropical Forest Science. Thanks are also due to Paolo Segre and Juan Di Trani for assistance in the field.

References

Condit R (1998). *Tropical Forest Census Plots*. Springer-Verlag and R. G. Landes Company, Berlin, Germany, and Georgetown, Texas.

Freedman D, Lane D (1983). "A Nonstochastic Interpretation of Reported Significance Levels." *Journal of Business and Economic Statistics*, **1**(4), 292–298. doi:10.2307/1391660.

Hubbell SP, Condit R, Foster RB (2005). "Barro Colorado Forest Census Plot Data." *URL https://ctfs.arnarb.harvard.edu/webatlas/datasets/bci*.

Hubbell SP, Foster RB, O'Brien ST, Harms KE, Condit R, Wechsler B, Wright SJ, Loo de Lao S (1999). "Light Gap Disturbances, Recruitment Limitation, and Tree Diversity in a Neotropical Forest." *Science*, **283**, 554–557.

Mrkvička T, Roskovec T, Rost M (2021). "A Nonparametric Graphical Tests of Significance in Functional GLM." *Methodology and Computing in Applied Probability*, **23**, 593–612. doi:10.1007/s11009-019-09756-y.

Mrkvička T, Myllymäki M (2023). "False Discovery Rate Envelopes." Statistics and Computing, **33**, 109. URL https://doi.org/10.1007/s11222-023-10275-7.

Nagy S, Gijbels I, Hlubinka D (2017). "Depth-Based Recognition of Shape Outlying Functions." *Journal of Computational and Graphical Statistics*, **26**(4), 883–893. doi:10.1080/10618600.2017.1336445.

Pedersen TL (2020). ***patchwork**: The Composer of Plots*. R package version 1.1.0, URL https://CRAN.R-project.org/package=patchwork.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Viladomat J, Mazumder R, McInturff A, McCauley DJ, Hastie T (2014). "Assessing the Significance of Global and Local Correlations Under Spatial Autocorrelation: a Nonparametric Approach." *Biometrics*, **70**(2), 409–418.

Wickham H (2016). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag, New York. ISBN 978-3-319-24277-4.

A. Preparation of GDP data

The GDP data are publicly available at
https://data.worldbank.org/indicator/NY.GDP.PCAP.CD.
The excel file that we downloaded was called
`API_NY.GDP.PCAP.CD_DS2_en_excel_v2_3358980.xls`.
The inflation rates are publicly available at
https://data.worldbank.org/indicator/NY.GDP.DEFL.KD.ZG.
The excel file that we downloaded was called
`API_NY.GDP.DEFL.KD.ZG_DS2_en_excel_v2_3469555.xls`,
from there we took only the inflation rates for United States. Both are distributed under the CC-BY 4.0 license (see https://datacatalog.worldbank.org/public-licenses#cc-by).

```
R> GDP <- read_excel("API_NY.GDP.PCAP.CD_DS2_en_excel_v2_3358980.xls",
+                    sheet=1, skip=3) #, detectDates=TRUE)
R> Inflation <- read_excel("API_NY.GDP.DEFL.KD.ZG_DS2_en_excel_v2_3469555.xls",
+                           sheet=1, skip=3)
R> Inflation <- data.frame(Year=names(Inflation)[-(1:5)],
+        Infl=as.numeric(Inflation[Inflation$`Country Code` == "USA", -(1:5)]))
R> Inflation$Infl <- Inflation$Infl/100 + 1
R> range(Inflation$Infl)
```

Then we discounted the GDP of every country in the study to the 1960 USD, and we extrapolated the missing values of the GDP of a country using the closest known ratio of the GDP of the country and the median GDP in that year. Further, the missing values of GDP were interpolated using linear interpolation of the two closest ratios.

```
R> GDPfuncs <- GDP[, 5:65]
R> GDPfuncs <- as.matrix(GDPfuncs)
R> rownames(GDPfuncs) <- GDP$`Country Name`
R> rownames(GDPfuncs)[rownames(GDPfuncs) == "United States"] <- "United States of America"
R> # Choose the countries which have more than 1 million,
R> # and that we selected above/here for population growth data
R> data("popgrowthmillion")
R> POPcset <- create_curve_set(list(r=1950:2014,
+                obs=popgrowthmillion[, -(132:134)])) # Left out Oceania
R> GDPfuncs <- GDPfuncs[rownames(GDPfuncs) %in% colnames(POPcset$funcs), ]
R> GDPcset <- create_curve_set(list(r = 1960:2020, obs=t(GDPfuncs)),
+                              allfinite = FALSE)
R> # Discounting (NAs stay NA here)
R> for(j in 1:dim(GDPcset$funcs)[2]){
+    for(i in 2:dim(GDPcset$funcs)[1]){
+      Inflat <- prod(Inflation$Infl[1:i-1])
+      GDPcset$funcs[i,j] <- GDPcset$funcs[i,j]/Inflat
```

```
+    }
+  }
R> # Median
R> GDPmedian <- GET:::curve_set_median(GDPcset, na.rm=TRUE)
R> # Interpolation and extrapolation of the values for NA's
R> for(i in 1:dim(GDPcset$funcs)[2]){
+    j=1
+    while(j <= dim(GDPcset$funcs)[1]) {
+      if(is.na(GDPcset$funcs[j,i])) {
+        k = j # k = first value which has a non-NA value
+        while(k <= dim(GDPcset$funcs)[1] && is.na(GDPcset$funcs[k,i])) {
+          k = k+1
+        }
+        if(j==1) { # j = Year. If j = 1, then missing values in the beginning
+          # Calculate the ratio to the median at Year k
+          ratio <- GDPcset$funcs[k,i]/GDPmedian[k]
+          for(l in j:(k-1)) {
+            GDPcset$funcs[l,i] = GDPmedian[l] * ratio
+          }
+        }
+        if(k>dim(GDPcset$funcs)[1]) { # The case with NA's at the end
+          # Calculate the ratio to the median at Year j-1
+          ratio <- GDPcset$funcs[j-1,i]/GDPmedian[j-1]
+          for(l in j:(k-1)) {
+            GDPcset$funcs[l,i] = GDPmedian[l] * ratio
+          }
+        }
+        if(j>1 && k<=dim(GDPcset$funcs)[1]) {
+          ratio.jm1 <- GDPcset$funcs[j-1,i]/GDPmedian[j-1] # start
+          ratio.k <- GDPcset$funcs[k,i]/GDPmedian[k] # end
+          for(l in j:(k-1)) {
+            GDPcset$funcs[l,i] = GDPmedian[l] *
+              ( ratio.jm1+(ratio.k-ratio.jm1)/(k-j+1)*(l-j+1) )
+          }
+        }
+      }
+      j=j+1
+    }
+  }
```

Let us then consider only those countries that exist in both data and arrange the data sets similarly.

```
R> #- Reduce POPcset to those countries that exist in GDP data
R> cond <- colnames(POPcset$funcs) %in% rownames(GDPfuncs)
R> POPcset <- subset(POPcset, cond)
R> #groups <- groups[cond]
```

```
R>
R> # Arrange the GDPcset to the same order as the POPcset and groups!
R> #-------------------------------------------------------------------
R> ids <- NULL
R> for(i in 1:ncol(POPcset$funcs))
+    ids[i] <- which(colnames(GDPcset$funcs) == colnames(POPcset$funcs)[i])
R> GDPcset$funcs <- GDPcset$funcs[,ids]
R> # check: colnames(GDPcset$funcs) == colnames(POPcset$funcs)
R>
R> # Reduce GDPcset and POPcset to those years that exist in both data
R> POPcset <- crop_curves(POPcset, r_min =1960, r_max=2014)
R> GDPcset <- crop_curves(GDPcset, r_min =1960, r_max=2014)
R>
R> #groups <- as.factor(groups)
```

This `GDPcset` was then saved as `GDP` data in **GET**.

Affiliation:

Mari Myllymäki
Natural Resources Institute Finland (Luke)
Latokartanonkaari 9
FI-00790 Helsinki, Finland
E-mail: mari.myllymaki@luke.fi
URL: https://www.luke.fi/en/experts/mari-myllymaki/